

Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

Claims 1 to 21 (cancelled).

Claim 22 (Currently amended): A control system for a device, wherein a graphic user interface (“GUI”) controller ~~operates a GUI of the device independently from~~ an embedded controller ~~from handling a GUI of the device of the device~~, the control system comprising:

the embedded controller for controlling and monitoring the device;

a liquid crystal display (“LCD”) for displaying the GUI to a user, the GUI including:

a first GUI object for displaying a status of the device; and

a second GUI object for displaying a command to the device;

a touch screen for detecting the command from the user;

the GUI controller, comprising:

at least one memory, comprising:

a document buffer storing a document (~~in HTML~~) defining an appearance of the GUI, the document comprising:

a first operation code (“opcode”) identifying the first GUI object;

a second opcode identifying the source of the status;

a third opcode identifying the second GUI object; and

a fourth opcode identifying the destination of the command;

a data buffer storing the status and the command;

a GUI object library storing:

a first set of executable codes defining an appearance and a functionality of the first GUI object with instructions for rendering the first GUI object, receiving the status from the embedded processor, and further rendering the first GUI object to show a visual response to the status;

a second set of executable codes defining an appearance and a functionality of the second GUI object with instructions for rendering the second GUI object, receiving the command from the touch screen, further rendering the second GUI object to show a visual response to the command, and sending the command to the embedded processor;

a frame buffer storing at least one complete display frame image of the GUI;

a GUI processor for rendering the GUI and handling user inputs independently from the embedded controller, wherein:

the GUI processor is coupled to the touch screen via a touch screen analog to digital converter to receive the command;

the GUI processor is coupled to the embedded processor via a serial UART interface to send the command and to receive the status;

the GUI processor is coupled to the at least one memory via a direct memory bus interface, wherein:

in response to the first and the second opcodes, the GUI processor executes the first set of executable codes to render the first GUI object to the frame buffer independently from the embedded processor, to communicate with the embedded

processor to receive the status from the embedded processor, and to further render the first GUI object to the frame buffer in response to the status independently from the embedded processor;

in response to the third and the fourth opcodes, the GUI processor executes the second set of executable codes to render the second GUI object to the frame buffer independently from the embedded processor, to receive the command from the touch screen independently from the embedded processor, to render the second GUI object again to the frame buffer to show a visual response to the command independently from the embedded processor, and to send the command to the embedded processor;

a pixel serializer coupled to the LCD to continuously refresh the LCD with the complete display frame image in the frame buffer that contains both the rendered first GUI object and the rendered second GUI object.

Claim 23 (Currently amended): A first controller providing a graphical user interface (“GUI”) for a device independently from a second controller that is monitored and controlled by a second controller monitors and controls the device, the first controller comprising:

at least one memory comprising:

a document buffer storing a document, the document comprising an operation code (“opcode”) identifying a GUI object in the GUI and a second opcode identifying a parameter of the GUI object, the parameter being from a source external to the first controller;

a frame buffer for storing at least one complete frame of the GUI including a visual representation of the GUI object;

a GUI object library storing executable codes defining an appearance and a functionality of the GUI object with instructions for communicating the

parameter with the source and rendering the GUI object in response to the parameter;

a processor coupled to the at least one memory, wherein the processor reads the first and the second opcodes, reads the executable codes, executes the executable codes to communicate the parameter with the source and to render the GUI object in response to the parameter independently from the second controller, and saves the rendered GUI object in the frame buffer.

Claim 24 (previously presented): The first controller of claim 23, further comprising an output device coupled to the frame buffer to receive the GUI, the output device displaying the GUI to the user.

Claim 25 (previously presented): The first controller of claim 24, wherein the output device is a liquid crystal display ("LCD").

Claim 26 (previously presented): The first controller of claim 25, further comprising a pixel serializer coupled between the frame buffer and the LCD, the pixel serializer outputting each line of the GUI in the frame buffer to the LCD.

Claim 27 (previously presented): The first controller of claim 23, wherein the source external to the first controller is an input device, the parameter is a command from the user to the second controller for controlling the device, and the processor communicates the parameter with the source by receiving the command from the input device.

Claim 28 (previously presented): The first controller of claim 27, wherein the executable codes further comprises the instructions for the processor to send the command to the second controller, the method further comprising the processor sending the command to the second controller.

Claim 29 (previously presented): The first controller of claim 28, wherein the input device is one of a touch screen, a key pad, an infrared remote, and a voice decoder.

Claim 30 (previously presented): The first controller of claim 29, wherein the GUI object is one of a button and a list.

Claim 31 (previously presented): The first controller of claim 23, wherein the source external to the first controller is the second controller, the parameter is a status of the device from the second controller to the user, and the processor communicates the parameter with the source by receiving the status from the second controller.

Claim 32 (previously presented): The first controller of claim 31, wherein the GUI object is a text field.

Claim 33 (previously presented): The first controller of claim 23, further comprising another memory coupled to the processor, the another memory storing the document, the processor buffering the document from the another memory to the at least one memory.

Claim 34 (previously presented): The first controller of claim 23, wherein the second controller further comprises another memory storing the document, the second controller reading the document from the another memory and sending the document to the first controller, the first controller storing the document in the at least one memory.

Claim 35 (currently amended): A method for a first controller to generate a graphic user interface ("GUI") for a device independently from a second controller that is ~~monitored and controlled by a second controller~~ monitors and controls the device, comprising:

writing a document in a document buffer in at least one memory, wherein the document defines an appearance of the GUI, the document comprising a first operation code ("opcode") identifying a GUI object and a second opcode identifying a parameter of the GUI object, the parameter being from a source external to the first controller;

in response to the first and the second opcodes, retrieving executable codes of the GUI object from a GUI object library stored in the at least one memory, wherein the executable codes comprises instructions that define an appearance and a functionality of the GUI object with instructions for communicating the parameter with the source and rendering the GUI object in response to the parameter;

independently from a second controller, executing the instructions to communicate the parameter with the source and to render the GUI object in response to the parameter;

writing the rendered GUI object in a frame buffer in the at least one memory; and
sending the rendered GUI object from the frame buffer to an output device, wherein
the output device displays the GUI to a user.

Claim 36 (previously presented): The method of claim 35, wherein the output device is a liquid crystal display ("LCD").

Claim 37 (currently amended): The method of claim 35, wherein the source external to the first controller is an input device, the parameter is a command from the user to the second controller for controlling the device, and the ~~processor~~ first controller communicates the parameter with the source by receiving the command from the input device.

Claim 38 (previously presented): The method of claim 37, wherein the executable codes further comprises the instructions for the first controller to send the command to the second controller, the method further comprising the first controller sending the command to the second controller.

Claim 39 (previously presented): The method of claim 38, wherein the input device is one of a touch screen, a key pad, an infrared remote, and a voice decoder.

Claim 40 (previously presented): The method of claim 39, wherein said rendering the GUI object comprises drawing one of a button and a list.

Claim 41 (currently amended): The method of claim 35, wherein the source external to the first controller is the second controller, the parameter is a status of the device from the second controller to the user, and the ~~processor~~ the first controller communicates the parameter with the source by receiving the status from the second controller.

Claim 42 (previously presented): The method of claim 41, wherein said rendering the GUI object comprises drawing a text field.

Claim 43 (previously presented): The method of claim 35, further comprising reading the document from another memory and buffering the document in the at least one memory.

Claim 44 (previously presented): The method of claim 35, further comprising receiving the document from the second controller and buffering the document in the at least one memory.

Claim 45 (withdrawn): A method for programming a graphic user interface ("GUI") for a GUI controller of a device, comprising:

creating a HTML page defining an appearance of the GUI, said creating comprising:

writing an applet code for a GUI object;

writing a reference to a source of a parameter of the GUI object;

compiling the HTML page into a compact HTML page, said compiling comprising

replacing the applet code with a first operation code ("opcode") identifying the GUI object;

replacing the reference with a second opcode identifying the parameter of the GUI object;

reading the compact HTML page from at least one memory in the device; and

in response to the first and the second opcodes, executing executable codes instructions in the at least one memory to communicate the parameter with the source and to render the GUI object in response to the parameter.

Claim 46 (withdrawn): The method of claim 45, wherein said communicating the parameter with the source comprises receiving a status of the device from an embedded controller, the embedded controller monitoring and controlling the device.

Claim 47 (withdrawn): The method of claim 45, wherein said communicating the parameter with the source comprises receiving a command from an input device.

Claim 48 (withdrawn): The method of claim 47, further comprising executing the executable codes to send the command to the embedded controller.